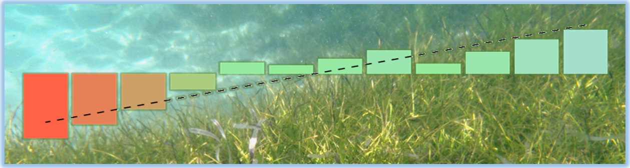
SWMPr: An R package for estuarine water quality time series

Marcus W. Beck[1]

[1]ORISE, USEPA NHEERL Gulf Ecology Division
Email: beck.marcus@epa.gov

April 15, 2015



Overview

A mixture of things...

- What is NERSS/SWMP and motivation for creating the package
- The process of package development
- What can SWMPr do
- What has SWMPr done

What is NERRS/SWMP?

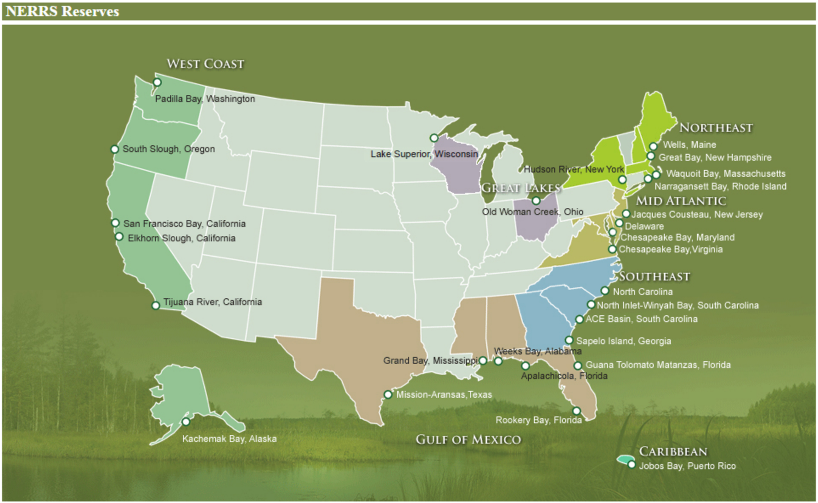
NERRS

National Estuarine Research Reserve System, established by Coastal Zone Management Act of 1972. Address goals for ***long-term research***, ***monitoring***, ***education***, and ***stewardship*** for more effective coastal management.

SWMP

System Wide Monitoring Program, initiated in 1995 to provide ***continuous monitoring*** data at over 300 stations in in each of the 28 NERRS reserves

What is NERRS/SWMP?



http://nerrs.noaa.gov/ReservesMap.aspx

What is NERRS/SWMP?

Each reserve has fixed, continuous monitoring stations for ***water quality*** (15 min), ***meteorology*** (15 min), and ***nutrients*** (monthly)

The parameters for a station are specific to the parameter type

Water quality	***Meteorology***	***Nutrients***
temp, spcond, sal, do_pct, do_mgl, depth, cdepth, level, clevel, ph, turb, chlfluor	atemp, rh, bp, wspd, maxwspd, wdir, sdwdir, totpar, totprcp, cumprcp, totsorad	po4f, chla_n, no3f, no2f, nh4f, no23f, ke_n, urea

What is NERRS/SWMP?

Data maintained by the Centralized Data Management Office (CDMO)

What is NERRS/SWMP?

As of April 10, > 58 million SWMP data records available from CDMO

Raw data will look like this...

	A	B	C	D	E	F	G	H	I	J	K	L
1	StationCo	isSWMP	DateTimeStamp	Historical	Provisiona	CollMetho	REP	F_Record	PO4F	F_PO4F	NH4F	F_NH4F
2	apacpnut	P	1/10/2012 10:20	0	1	1	1		0.003	<-4> [SBL]	0.03	<0>
3	apacpnut	P	2/7/2012 11:41	0	1	1	1		0.005	<0>	0.019	<0>
4	apacpnut	P	3/5/2012 11:51	0	1	1	1		0.003	<-4> [SBL]	0.041	<0>
5	apacpnut	P	4/4/2012 10:30	0	1	1	1		0.003	<-4> [SBL]	0.043	<0>
6	apacpnut	P	5/9/2012 10:12	0	1	1	1		0.003	<0>	0.053	<0>
7	apacpnut	P	5/9/2012 10:15	0	1	1	2		0.003	<-4> [SBL]	0.022	<0>
8	apacpnut	P	5/9/2012 10:20	0	1	1	3		0.003	<0>	0.016	<0>
9	apacpnut	P	6/5/2012 8:30	0	1	1	1		0.003	<-4> [SBL]	0.04	<0>
10	apacpnut	P	7/3/2012 9:58	0	1	1	1	{CSM}	0.004	<0>	0.094	<0>
11	apacpnut	P	7/3/2012 9:59	0	1	1	2	{CSM}	0.004	<0>	0.066	<0>
12	apacpnut	P	7/3/2012 10:01	0	1	1	3	{CSM}	0.005	<0>	0.069	<0>
13	apacpnut	P	8/7/2012 9:53	0	1	1	1	{CSM}	0.003	<-4> [SBL]	0.05	<0>
14	apacpnut	P	9/5/2012 10:56	0	1	1	1		0.003	<-4> [SBL]	0.026	<0>
15	apacpnut	P	10/2/2012 9:22	0	1	1	1		0.003	<-4> [SBL]	0.042	<0>
16	apacpnut	P	10/2/2012 9:27	0	1	1	2		0.003	<-4> [SBL]	0.024	<0>
17	apacpnut	P	10/2/2012 9:32	0	1	1	3		0.003	<0>	0.042	<0>
18	apacpnut	P	11/6/2012 10:30	0	1	1	1		0.003	<-4> [SBL]	0.07	<0>
19	apacpnut	P	11/26/2012 11:39	0	1	1	1		0.003	<-4> [SBL]	0.041	<0>

What is the problem?

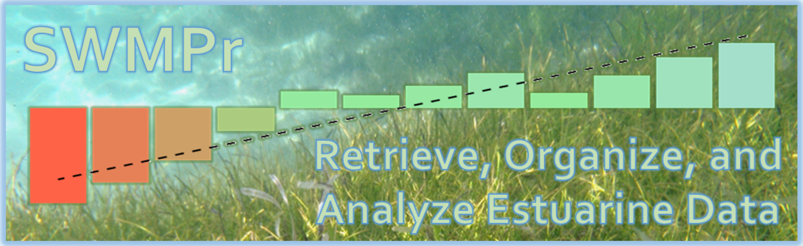
An invaluable data source but no recent comparative analyses between systems

NERRS researchers, managers, and technicians need more tools for trend analysis

Some specific issues:

- Knowing what data to use and how to obtain
- Dealing with QAQC columns or removing ‘bad’ observations
- Combining data for comparison
- Issues inherent with time series, e.g., signal vs. noise, data quantity

What is the (potential) solution?



SWMPr v2.0.0 is now available on CRAN!

```
> install.packages('SWMPr')
> library(SWMPR)
```

Currently no vignette, but working on a manuscript

The process of package development

The development version lives on GitHub:
https://github.com/fawda123/SWMPr

The package development process was much simplified using RStudio and the Hadleyverse (specifically devtools, roxygen2)

In RStudio, create a package template:

`File` > `New Project` > `New Directory` > `R Package`, with options for Git version control

Package does not have to be on CRAN to distribute...

The process of package development

Follow the advice here: http://r-pkgs.had.co.nz/

R packages by Hadley Wickham

Table of contents

Want to learn from me in person? I'm next teaching in Chicago, May 27-28.

Want a physical copy of this material? Buy from amazon!.

Contents

How to contribute

Edit this page

R packages

This is the in-progress book site for **"R packages"**. It will be published with O'Reilly around March 2015. You can pre-order a copy from amazon.

Packages are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data. In this section you'll learn how to turn your code into packages that others can easily download and use. Writing a package can seem overwhelming at first. So start with the basics and improve it over time. It doesn't matter if your first version isn't perfect as long as the next version is better.

Getting started

- Introduction
- Package structure

What can SWMPr do?

SWMPr functions are grouped into three categories that describe their use in the 'data workflow'

Retrieve	***Organize***	***Analyze***
`all_params`	`comb`	`aggreswmp`
`all_params_dtrng`	`qaqc`	`aggremetab`
`import_local`	`qaqcchk`	`ecometab`
`import_remote`	`rem_reps`	`decomp`
`single_param`	`setstep`	`decomp_cj`
`site_codes`	`subset`	`hist`
`site_codes_ind`		`lines`
		`na.approx`
		`plot`
		`plot_metab`
		`plot_summary`
		`smoother`

How are data ***retrieved***?

The first challenge is to determine the station, parameter, and date range of interest - Check the available data on the CDMO website

Also familiarize yourself with the NERRS/SWMP naming convention

Site (reserve), ***station***, and ***parameter type*** are identified by a 7 or 8 character name

E.g., elkcwmet

- elk: site, Elkhorn Slough
- cw: station, Caspian Weather Station
- met: parameter type (weather)

How are data ***retrieved***?

SWMP data can be imported from CDMO into R three ways

1) From a local path: `import_local`

Advantages:

- Ideal for large datasets
- Most recent data

Disadvantages:

- Outside of R
- Only works for one type of data request from CDMO

How are data ***retrieved***?

SWMP data can be imported from CDMO into R three ways

2) retrieve SWMP data from a third-party server: `import_remote`,

Advantages:

- Fast import!
- No requests to CDMO

Disadvantages:

- Data are not current
- Requires further processing - date subsets, etc.

How are data ***retrieved***?

SWMP data can be imported from CDMO into R three ways

3) Use the existing CDMO web services to import directly:
`single_param`, `all_params`, `all_params_dtrng`

Advantages:

- Current
- Customized requests

Disadvantages:

- IP address must be registered
- Quantity severely limited

How are data *retrieved*?

The best approach depends on your needs

The end result is the same - data are imported as a `swmpr` S3 object

```
> dat <- import_remote('kacsswq')
> class(dat)

## [1] "swmpr"      "data.frame"

> names(attributes(dat))

## [1] "names"       "row.names"   "class"       "station"
## [5] "parameters"  "qaqc_cols"   "date_rng"    "timezone"
## [9] "stamp_class"
```

How are data *retrieved*?

The remaining functions have `swmpr` methods

```
> methods(class = 'swmpr')

##  [1] aggremetab.swmpr*   aggreswmp.swmpr*
##  [3] comb.swmpr*         decomp.swmpr*
##  [5] decomp_cj.swmpr*    ecometab.swmpr*
##  [7] hist.swmpr*         lines.swmpr*
##  [9] na.approx.swmpr*    plot.swmpr*
## [11] plot_metab.swmpr*   plot_summary.swmpr*
## [13] qaqc.swmpr*         qaqcchk.swmpr*
## [15] rem_reps.swmpr*     setstep.swmpr*
## [17] smoother.swmpr*     subset.swmpr*
##
##    Non-visible functions are asterisked
```

`swmpr` objects also inherit methods from the `data.frame` class

How are data ***organized***?

Data organization depends on the analysis needs - it is neither fun nor straightforward (common opinion, not mine)

What are some challenges?

- Imported data have QAQC columns
- Extra columns/rows
- Maybe we don't care about all the parameters
- Data from separate sites are in separate objects

The ***organize*** functions are specific to the SWMP data but many of the principles apply to generic time series

How are data ***organized***?

A relevant example - we want to compare time series from different sites

- Data may have arbitrary time steps that do not match between sites
- Date ranges may also differ

The `setstep` and `comb` functions address these issues!

```
> met <- import_remote('apaebmet')
> wq <- import_remote('apacpwq')
> dat <- comb(met, wq) # tada!
```

How are data ***organized***?

The `setstep` function is used within `comb` to standardize the time steps for each input object

A tricky problem - actual observations which may occur on an arbitrary step must be matched to a set time step

This function uses 'fast-ordered joins' from the data.table package using the 'nearest' method

Also must define a threshold for matching: +/- some buffer of allowance beyond which matches are discarded

How are data ***organized***?

Mechanistically, `setstep` does the following for each data object:

- Create a continuous 'master' time series at defined step using first/last time stamps
- Match existing observations to standardized using 'nearest' join method
- Calculate difference in time between matched and standardized step, discard those beyond threshold

Standardized datasets are then combined by absolute matching of time steps

How are data *organized*?

```
> dim(met)

## [1] 490847     11

> dim(wq)

## [1] 455808     13

> # standardize time step to two hours
> # maximum difference for matching 30 minutes
> # combine only overlapping time ranges
> dat <- comb(wq, met, timestep = 120, differ = 30,
+   method = 'intersect')
> dim(dat)

## [1] 56977    23
```

How are data ***organized***?

```
> head(dat, 4)

##          datetimestamp atemp rh   bp wspd maxwspd wdir
## 1 2001-12-31 23:00:00     4 69 1017    4      NA  347
## 2 2002-01-01 01:00:00     3 75 1017    3      NA    9
## 3 2002-01-01 03:00:00     2 77 1018    3      NA  331
## 4 2002-01-01 05:00:00     1 82 1019    4      NA    0
##   sdwdir totpar totprcp totsorad temp spcond sal do_pct
## 1     NA      0      NA       NA   NA     NA  NA     NA
## 2     NA      0      NA       NA   12     37  24    104
## 3     NA      0      NA       NA   12     40  26     99
## 4     NA      0      NA       NA   11     42  26     98
##   do_mgl depth cdepth level clevel ph turb chlfluor
## 1     NA    NA     NA    NA     NA NA   NA       NA
## 2     10     2     NA    NA     NA NA    3       NA
## 3      9     2     NA    NA     NA NA    4       NA
## 4      9     2     NA    NA     NA NA    5       NA
```

How are data ***analyzed***?

Time series analysis can be very general or very specific...

Applications